

## INFORMATION PROCESSOR

Publication number: JP6044129

Publication date: 1994-02-18

Inventor: SAKAKIBARA MASAHITO

Applicant: FUJI XEROX CO LTD

Classification:

- international: G06F11/30; G06F12/02; G06F12/16; G06F11/30;  
G06F12/02; G06F12/16; (IPC1-7): G06F12/02;  
G06F11/30; G06F12/16

- European:

Application number: JP19920218388 19920724

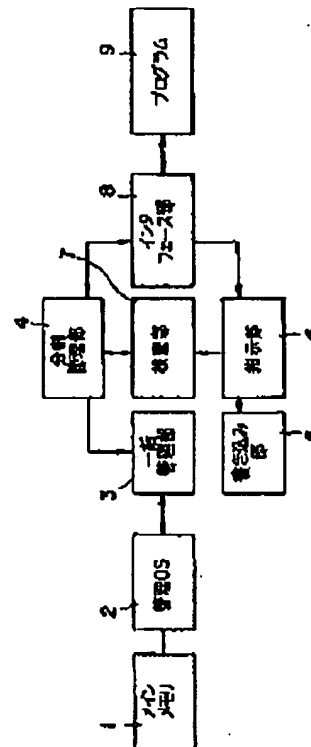
Priority number(s): JP19920218388 19920724

Report a data error here

### Abstract of JP6044129

**PURPOSE:** To provide an information processor where the deterioration of processing capacity on account of overhead does not occur and the release of a memory area can easily be executed or the detection of illegal memory access is easy.

**CONSTITUTION:** The memory areas 11 in various sizes, which a program 9 requires, are collectively obtained from an operating system 2 by a first memory area management means 3. The collectively obtained areas are managed by dividing them by a second memory area management means 4. At the time of executing the program, the areas are transferred with the program. When one job terminates, the areas are collectively returned to the operating system 2. At the time of debugging, a write means 5 writes a data pattern into a prescribed part when the areas are transferred between the second memory area management means 4 and the program. An inspection means 7 inspects the presence or absence of the destruction.



Data supplied from the *esp@cenet* database - Worldwide

01-29-'07 16:33 FROM-Mattingly, Stanger

703-684-1157

T-239 P019/035 F-870

<http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=JP6044129&F=0>

1/24/2007



(2)

特開平6-44129

1

2

## 【特許請求の範囲】

【請求項1】 プログラム実行時の一時記憶装置となるメモリと、

該メモリを管理するオペレーティングシステムと、  
前記プログラムに代わり前記オペレーティングシステムとの間で前記メモリの領域の授受を行なう第一のメモリ領域管理手段と、

該手段により取得されたメモリ領域を管理し、前記プログラムとの間で、該管理しているメモリ領域の授受を行なう第二のメモリ領域管理手段とを備えたことを特徴とする情報処理装置。

【請求項2】 プログラム実行時の一時記憶装置となるメモリと、

該メモリを管理するオペレーティングシステムと、  
前記プログラムに代わり前記オペレーティングシステムとの間で前記メモリの領域の授受を行なう第一のメモリ領域管理手段と、

該手段により取得されたメモリ領域を管理し、前記プログラムとの間で、該管理しているメモリ領域の授受を行なう第二のメモリ領域管理手段と、

デバッグ時の動作をするか実行時の動作をするかを指示する指示手段と、

該手段によりデバッグ時の動作が指示されたとき、前記プログラムとの間のメモリ領域の授受に際し、該領域に係る所定部分に任意のデータパターンを書き込む書き込み手段と、

該データパターンの破壊の有無を検査する検査手段とを備えたことを特徴とする情報処理装置。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は情報処理装置に関し、詳しくは情報処理装置におけるメモリ領域の管理方式に関する。

【0002】

【従来の技術】 従来、プログラム実行時の一時記憶装置のメモリ領域の取得、解放にはオペレーティングシステムの提供するシステムコールや標準ライブラリが使用されていた（例えば特開平2-287630、Sun Microsystems, Inc. 編 "SunOS Reference Manual" (1988) Sun Microsystems, Inc. (米) 等参照。）。(なお以下の説明では「一時記憶装置のメモリ領域」を単に「メモリ領域」と言う。)

【0003】

【発明が解決しようとする課題】 上記システムコールや標準ライブラリを使用したメモリ領域の取得と解放には、天々に所定のオーバーヘッドタイムを必要とする。このためメモリ領域の取得と解放とが頻繁に繰り返されるプログラムの場合、その積算時間はかなりのものとなり、オーバーヘッドによる処理性能の低下を招いていた。また取得した領域は、資源の無駄遣いにならないよ

う、例えば一つのジョブが終了した時点等にきちんと解放しておかなければならない。しかし、メモリ領域の取得、解放が頻繁に行なわれるプログラムに於て、これらを忘れずにきちんと実行するようにプログラミングすることは、プログラマにとって負担である。更にこれに関するバグとして、取得領域以外への不正アクセスがある。しかしチェックをしても自分自身のミスにはなかなか気が付かないのが世の常であり、仮に領域の取得と解放の回数が少なかったとしても、その発見はそれ程容易なことではない。ましてや領域の取得と解放が頻繁に行なわれていた場合、その発見はなかなか難しい。

【0004】 本発明の第一の目的は、メモリ領域の取得と解放が頻繁に行なわれても、オーバーヘッドによる処理性能の低下が起こらない情報処理装置を提供することにある。本発明の第二の目的は、メモリ領域の取得、解放に関しプログラミングの際の負担が少なく、資源の無駄遣いが起こりにくい情報処理装置を提供することにある。本発明の第三の目的は、不正なメモリアクセスの発見が容易な情報処理装置を提供することにある。

【0005】

【課題を解決するための手段】 上記第一及び第二の目的達成のため請求項1の発明は、プログラム実行時の一時記憶装置となるメモリと、該メモリを管理するオペレーティングシステムと、前記プログラムに代わり前記オペレーティングシステムとの間で前記メモリの領域の授受を行なう第一のメモリ領域管理手段と、該手段により取得されたメモリ領域を管理し、前記プログラムとの間で、該管理しているメモリ領域の授受を行なう第二のメモリ領域管理手段とを備える。また上記第三の目的達成のため請求項2の発明は、プログラム実行時の一時記憶装置となるメモリと、該メモリを管理するオペレーティングシステムと、前記プログラムに代わり前記オペレーティングシステムとの間で前記メモリの領域の授受を行なう第一のメモリ領域管理手段と、該手段により取得されたメモリ領域を管理し、前記プログラムとの間で、該管理しているメモリ領域の授受を行なう第二のメモリ領域管理手段と、デバッグ時の動作をするか実行時の動作をするかを指示する指示手段と、該手段によりデバッグ時の動作が指示されたとき、前記プログラムとの間のメモリ領域の授受に際し、該領域に係る所定部分に任意のデータパターンを書き込む書き込み手段と、該データパターンの破壊の有無を検査する検査手段とを備える。

【0006】

【作用】 請求項1の発明に於て、第一のメモリ領域管理手段は、実行されるプログラムに代わり、プログラムが要求するサイズのメモリ領域をオペレーティングシステムから一括して取得する。第二のメモリ領域管理手段は、一括取得したメモリ領域を、プログラムの要求に対応したサイズに分割して管理し、プログラム実行時、該プログラムが要求するサイズのメモリ領域を、該プロ

(3)

特開平6-44129

3

ラムに引き渡す。また該プログラムからメモリ領域の返却があったときは、これを受け取り、オペレーティングシステムには返さず、自己の管理するツリー等の中に戻す。そして又要求があったとき、これらツリーの中から、要求されたサイズに合うものを取り出して該プログラムに引き渡す。即ちこの請求項1の発明では、プログラム実行時のメモリ領域の授受が、該プログラムと第二のメモリ管理手段との間で行なわれる。従って、従来装置のように、一回一回、オペレーティングシステムとの間でヒープの要求、返却をすることによるオーバーヘッドが無く、メモリ領域の授受が頻繁に実行されたとしても、オーバーヘッドによる処理性能の低下を来さない。

【0007】また請求項1の発明に於て、第一のメモリ管理手段は、例えば一つのジョブが終了した時点で、アプリケーションプログラムからの要求に応動し、第二のメモリ管理手段が分割管理していたメモリ領域を一括してオペレーティングシステムに返却する。即ちこの請求項1の発明では、メモリ領域の解放が、前記オペレーティングシステムから取得したメモリ領域、即ち分割管理されていたメモリ領域全体について、一括して行なわれる。従って、メモリの取得、解放が頻繁に行なわれるプログラムであっても、例えば一つのジョブの終了時に、この一括解放を一回実行することにより、メモリ領域の解放を確実に行なうことが出来、プログラムはメモリ領域の取得と解放の対応を全く気にせずにプログラミングを行なうことが出来る。また資源の無駄遣いも確実に防止される。

【0008】請求項2の発明に於て、マウスやキーボード等の指示手段により、或いは起動時の初期設定、環境設定等により、デバッグ時の動作が指示されると、書き込み手段は、前記プログラムとの間で行なわれるメモリ領域の授受に際し、該領域に係る所定部分に適宜のデータパターンを書き込む。即ちプログラムへのメモリ領域引き渡しの際は、該領域及びその前後に設定されたマージン領域等に、夫々別の適宜のデータパターン、「a b a b a b ……」等を書き込む。またプログラムからのメモリ領域の返却の際は、例えば該返却された領域及びマージン領域等に、更に夫々別の適宜のデータパターン例えば「f 0 f 0 f 0 ……」というようなデータパターン等を書き込む。検査手段はデータパターンの破損が無いかな否かを検査する。例えば引き渡されたメモリ領域の回りのマージン領域、或いは返却された後の当該メモリ領域は、プログラムがアクセスする筈の無いメモリ領域である。従って例えばメモリ領域が返却された際、これら領域のデータ破損が生じていたら、該プログラムがメモリの不正アクセスをしていることになる。即ち請求項2の発明によれば従来発見困難であったこの種バグの発見が容易に出来るようになる。

【0009】

【実施例】以下本発明の詳細を図示実施例に基いて説明

4

する。実施例装置の構成を図1に示す。図に於て1はメインメモリで、プログラム9が実行される際の一時記憶装置となる。2はオペレーティングシステム（管理OS）で、前記メインメモリ1のメモリ領域を管理する。この管理は公知の手法で行なわれる。3は一括管理部で、プログラム9の起動時に該プログラム9から要求されたメモリ領域を、オペレーティングシステム2に一括要求する。オペレーティングシステム2は要求されたサイズのメモリ領域をメインメモリ1のメモリ領域の中から取り出し、該管理部3に引き渡す。また該管理部3からメモリ領域が返還されたとき、これを受け取り自己の管理下に置く。なおオペレーティングシステム2と一括管理部3との間で授受されるメモリ領域を「一括領域」という。4は分割管理部で、前記一括領域を分割して管理し、プログラム9から要求があると、その要求に合ったサイズのメモリ領域をこの一括領域から取り出して、該プログラム9に引き渡す。また該プログラム9からこのメモリ領域が返還されたときは、該返還されたメモリ領域をオペレーティングシステムには返さず、自己の管理下に置く。なお分割管理部4とプログラム9の間で授受されるメモリ領域を「分割領域」という。

【0010】5は書き込み部で、指示部6から、デバッグ時の動作が指示されているとき、前記分割領域の授受に際し、該領域に係る所定部分に任意のデータパターンを書き込む。即ち本実施例では、プログラムへの引き渡しの際は、図4に示すように、要求されたサイズのメモリ領域11（分割領域）とこれに付されるノードヘッダ12の前後に、接頭マージンエリア13及び接尾マージンエリア14が付加され、これら3つの各領域に、例えば16進数a bの連続から成る「a b a b a b ……」等の、3種類の異なるデータパターンが書き込まれる。またプログラムからこれらメモリ領域11～14が返却された際は、該返却された3つの各領域に、引き渡しの際とは別の3つの異なるデータパターン例えば「f 0 f 0 f 0 ……」等が書き込まれる。この例のように、引き渡すときと返却されたとき夫々で、異なる例えば6通りのデータパターンを書き込むようにしておくと、メモリの不正参照等に係るデータパターンの種類から、不正アクセスしているメモリ領域の特定が出来、その情報から、当該領域への不正参照等を行なっているステップの探索等が容易に出来るようになる。またオペレーティングシステムが使用しているメモリ領域へのアクセスは、オペレーティングシステムがエラーとしてはわる。そこで、これらオペレーティングシステムが使用する領域のアドレスをデータパターンに用い、不正参照があった場合、その結果として行なわれるオペレーティングシステム使用領域へのメモリアクセスを該オペレーティングシステムが検出するようにして、これらのバグの検出に役立てても良い。

【0011】指示部6は、キーボード及びマウスを備え

(4)

特開平6-44129

5

る。この指示部6によりプログラム9の実行形式、即ち通常に実行するかデバッグ形式で動作するかが指示される。なお装置起動時やプログラム起動時の環境設定、初期設定、オプションなどにより、何れの実行形式を採るかの指示をするようにしても良い。この場合、これら設定等が請求項にいう指示手段となる。7は検査部で、プログラム9がデバッグ形式で実行されるとき、前記書き込み部5によって書き込まれたデータパターンの破壊の有無を検査する。8はインタフェース部で、分割管理部4及び指示部6と、プログラム9との間の橋渡しを行なう。なおプログラム9は、オペレーティングシステム2が提供する環境で動くものであれば、アプリケーションプログラム、開発用プログラム、その他如何なるものでも構わない。なおこれら実施例の各部分と、請求項の各構成との対応は、符号の説明の項に示す。

【0012】図2及び図3も引用して、本実施例装置の動作を説明する。先ず従来装置に於て実行していたように、プログラム9はそのジョブの実行に必要なメモリ領域の空間サイズ、個数、初期サイズ、拡張サイズ等を宣言する。これら空間サイズ等は従来装置におけると同様に、その使用頻度、使用の形態に応じてプログラム9側の部合で定める。一括管理部3はこれに応動して、要求された大きさのメモリ領域（一括領域）をオペレーティングシステム2から取得する。取得された一括領域は分割管理部4に引き渡される。分割管理部4は、この一括領域をサイズ別に分けて分割領域とし、ツリー構造で管理する。なおこのツリーはいわばプログラムに渡されていないフリーのメモリ領域を接続するものであるので、「フリーツリー」と称す。フリーツリーの管理アルゴリズムは公知のもの、例えばSOSP83などを用いる。そして本実施例ではオーバーヘッドタイムを出来るだけ少なくする為、分割領域のサイズを、予め設定された固定長と、プログラムの要求に応じて定められる可変長の二通りとし、固定長のときは、サイズの処理を省略した高速の領域設定を実行する（ステップS1、S2、S3）。なお「ステップ」の語は以後省略する。

【0013】領域設定の処理が終了すると、プログラム9は目的のジョブ、例えば、ページ記述言語をイメージデータに展開する等の処理を実行する。この過程で、プログラム9は、従来オペレーティングシステムに対して行なっていたメモリ領域の要求を、分割管理部4に対して行なう。分割管理部4は、プログラム9からの要求がメモリ領域の要求であるとき（S4「取得」）、要求されたメモリ領域サイズを自己が管理している各分割領域のサイズと比較し、対応できるなら（S5「十分」）、管理下のフリーツリーから要求されたサイズの分割領域を取り出し、プログラム9に引き渡す（S6）。プログラム9は渡された分割領域を使用して各処理を実行する。上記各処理の過程で、プログラム9がメモリ領域の解放（返却）を行なったとする（S4「解放」）。分割

6

管理部4はプログラム9の実行中に返却された分割領域はオペレーティングシステム2には返却せず、自己の管理下のフリーツリーの当該サイズの枝に接続する。

【0014】これら分割領域の引き渡し、返却は、プログラム9の内容に応じて各処理の過程で実行される。分割管理部4は、その都度自己の管理下のフリーツリーからの分割領域の引き出し及び該フリーツリーへの接続を繰り返す。これにより各分割領域は繰り返し利用される。なおサイズが足りないとき（S5「不足」）、分割管理部4は、オペレーティングシステム2から拡張サイズ分のメモリ領域を取得して自己のフリーツリーに組み入れ（S8）、プログラム9に引き渡す（S6）。これにより従来装置のように、一回一回、オペレーティングシステムとの間でヒープの要求、返却をすることによるオーバーヘッドが発生しなくなり、メモリ領域の授受が頻繁に実行されたとしても、オーバーヘッドによる処理性能の低下が起こらなくなる。

【0015】分割管理され、繰り返し利用された分割領域は、一つのジョブの終了時等に、一括してオペレーティングシステム2に返却される。即ち、プログラム9は当該ジョブ終了時等に、自己が設定した一括領域の解放（返却）を宣言する。これに応動して、一括管理部3は当該一括領域を、いわば丸ごと一括してオペレーティングシステム2に返却する（S11）。これにより、従来装置に比べ、オーバーヘッドタイムが大幅に短縮される。またそれ迄に多数のメモリ領域を取得していても、この宣言一つで一括返却が出来るので、プログラミングが容易になる。また解放忘れによる資源の無駄遣い等も生じなくなる。

【0016】作成したプログラムのデバッグをしたい場合は、指示部6から、或いは前記プログラムの設定等によりデバッグ時の動作を指示する。この指示があると実施例装置は図2のS6に代え図4のS61を、また図2のS7に代え図5のS71～S74の処理を、夫々実行する。即ちプログラム9からメモリ領域が要求されたとき、分割管理部4は、マージンエリア11及び14を加えた大きさのメモリ領域を、管理下のフリーツリーから取り出す（S61）。そして各エリア11、13及び14に、夫々別個の3種類の適宜のデータパターン、例えば「a b a b……」の繰り返しから成るデータパターンを書き込み、プログラム9に引き渡す。

【0017】またプログラム9からメモリ領域が返還されたとき、検査部7は当該領域を始めフリーツリーの各メモリ領域のデータパターン（例えば前記「a b a b……」）が破壊されていないかどうかを検査する（S71）。データが変化していれば、作用の項で説明した通り、そのときのプログラム9は、本来使用しない筈の領域を使用するというバグを含んでいる。この場合は、不図示ディスプレイへの表示やブザー鳴動によるエラー告知が為される（S72、図6S21、S22）。データ

(5)

特開平 6-44129

7

破壊が無ければ(S71「正常」)、そのとき返却されたメモリ領域の各領域11、13及び14に、上記引き渡しのときは更に別の3種類の適宜のデータパターン、例えば「f0f0f0 ……」の繰り返しからなるデータパターンが書き込まれる(S73)。データパターンが書き込まれたメモリ領域は、分割管理部4が管理するフリーツリーに接続される。その後、処理は図2のS4に引き継がれる。

【0018】なお検査のためのデータパターンを書き込むメモリ領域は任意で、例えばマージンエリア13、14のみ等としても良い。

【0019】

【発明の効果】以上説明したように、請求項1の発明では、プログラム実行時のメモリ領域の授受が第二のメモリ管理手段とプログラムとの間で行なわれる。従って、従来装置のように、一回一回、オペレーティングシステムとの間でメモリ領域の授受をすることによるオーバーヘッドは発生せず、メモリ領域の授受が頻繁に実行されたとしても、オーバーヘッドによる処理性能の低下が起こらない。同じく請求項1の発明では、プログラム終了時のメモリ領域の返却(解放)が、分割管理されていたメモリ領域全体について一括して行なわれる。従って、一つのジョブの中で取得した多数のメモリ領域もこの処理一つで簡単に解放することが出来、プログラミングが簡単になる。また従来装置のようにメモリ領域の解放忘れによる資源の無駄遣い等も生じない。また請求項2の発明では、指示手段によりデバッグ時の動作が指示されると、書き込み手段が、プログラムとの間で行なわれる

8

メモリ領域の授受等の際に、マージン領域やプログラムが使用していた領域等に任意のデータパターンを書き込み、検査手段がその破壊の有無を検査する。従ってそのとき実行されたプログラムが、メモリ領域に対する不正アクセスをしていれば、前記検査により直ちに発見が出来る。即ち請求項2の発明によれば従来発見困難であったこの種バグの発見が容易に出来る。

【図面の簡単な説明】

【図1】実施例装置の構成を示すブロック図。

【図2】メモリ領域取得及び管理の手順の例を示すフローチャート。

【図3】メモリ領域の一括削除の手順を示すフローチャート。

【図4】デバッグ時のメモリ領域引き渡しの手順の例を示すフローチャート。

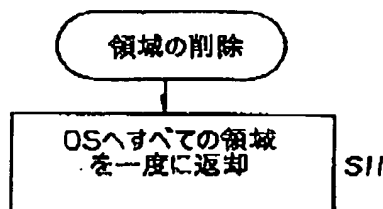
【図5】デバッグ時のメモリ領域返却の手順の例を示すフローチャート。

【図6】データパターンの検査手順の例を示すフローチャート。

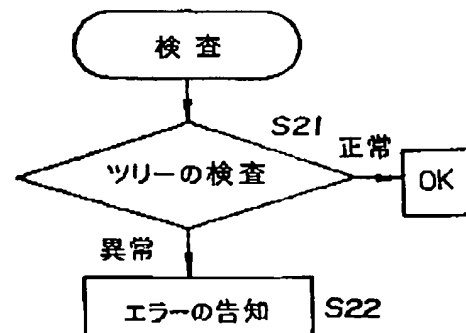
【符号の説明】

- 1 メモリ
- 2 オペレーティングシステム
- 3 第一のメモリ領域管理手段
- 4 第二のメモリ領域管理手段
- 5 書き込み手段
- 6 指示手段
- 7 検査手段
- 9 プログラム

【図3】



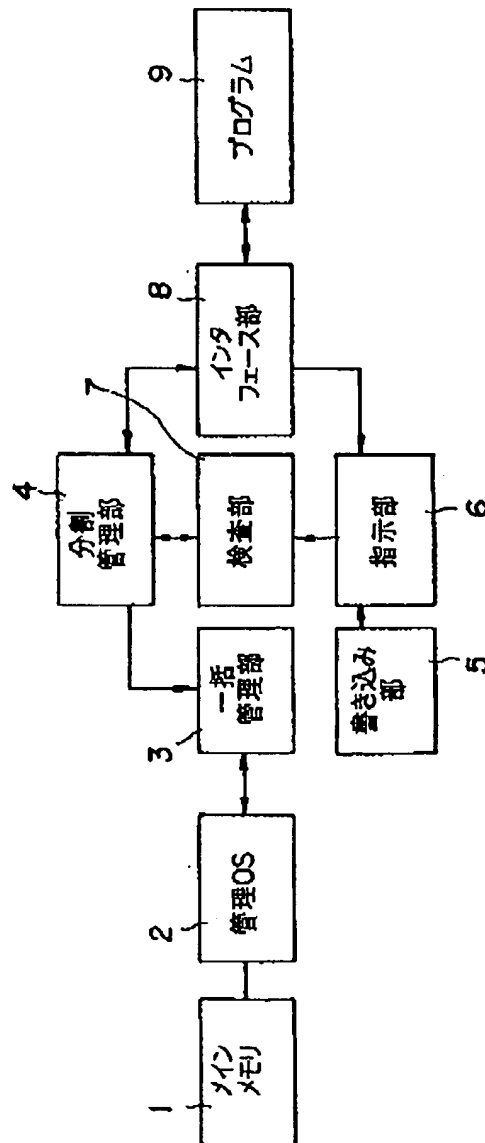
【図6】



(6)

特開平6-44129

【図1】

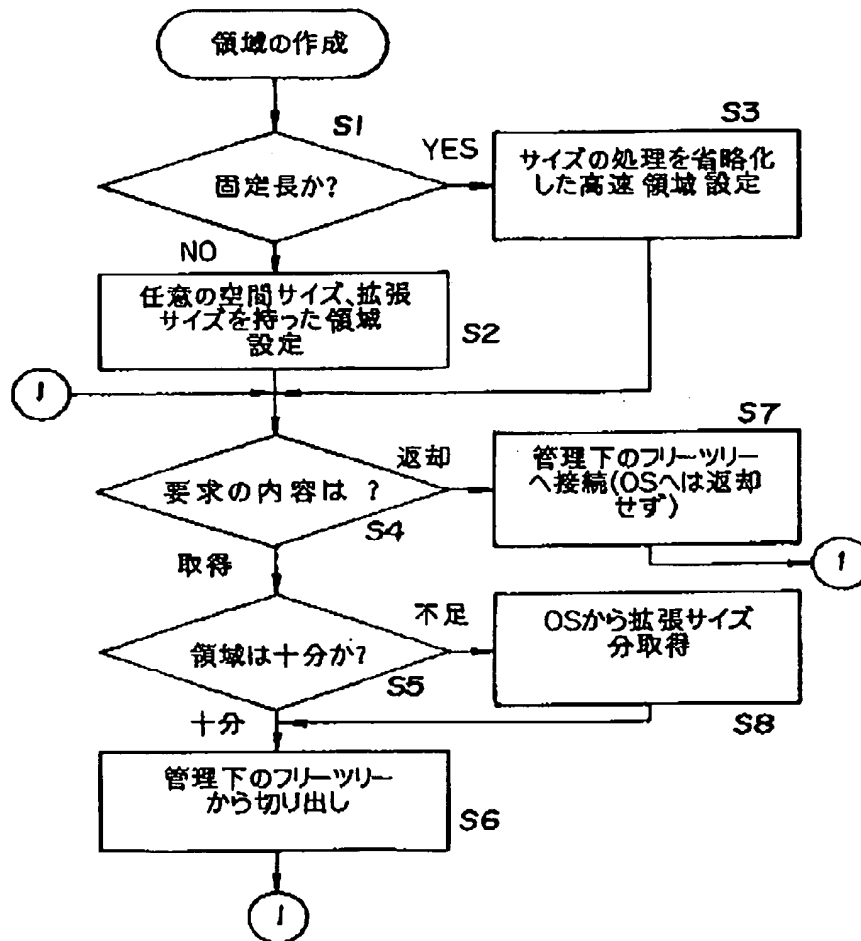




(7)

特開平6-44129

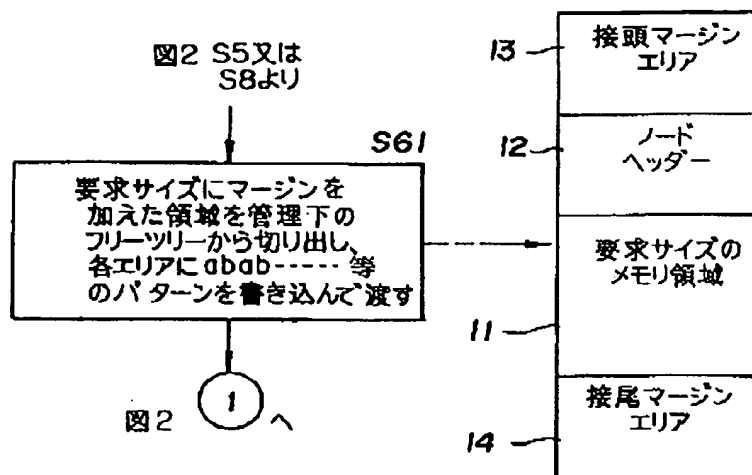
【図2】



(8)

特開平6-44129

【図4】



【図5】

